



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/696,828

10/30/2003

Joseph G. Laura

IDF 2506 (4000-14800)

1867

28003

7590

04/18/2007

SPRINT

6391 SPRINT PARKWAY

KSOPHT0101-Z2100

OVERLAND PARK, KS 66251-2100

EXAMINER

CHEN, QING

ART UNIT

PAPER NUMBER

2191

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
--	-----------	---------------

3 MONTHS

04/18/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary

Application No.

10/696,828

Applicant(s)

LAURA, JOSEPH G.

Examiner

Qing Chen

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 30 October 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-26 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-26 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 01 March 2004 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 20040209, 20050531.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____.

DETAILED ACTION

1. This is the initial Office action based on the application filed on October 30, 2003.
2. **Claims 1-26** are pending.

Drawings

3. The drawings were received on March 1, 2004. These drawings are not acceptable because the drawings are not in compliance with 37 CFR § 1.121(d). Any changes to an application drawing must be in compliance with 37 CFR § 1.84 and must be submitted on a replacement sheet of drawings, which shall be an attachment to the amendment document and, in the top margin, labeled "Replacement Sheet."

Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the Examiner, the Applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

4. The disclosure is objected to because of the following informalities: the application numbers for the U.S. patent applications incorporated by reference are missing on page 1, paragraph [001].

Appropriate correction is required.

Claim Objections

5. **Claims 6, 11, and 19** are objected to because of the following informalities:
- **Claim 6** contains a typographical error: the article "A" should be added before the statutory category of invention (*i.e.*, method).
 - **Claim 11** contains a typographical error: a period (.) should be added after the limitation body.
 - **Claim 19** contains a typographical error: "operation system" should read -- operating system --.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. **Claims 18-20** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 18 and 19 recite the limitation "the operating system." There is insufficient antecedent basis for this limitation in the claims. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading "an operating system" for the purpose of further examination.

Claim 20 depends on Claim 19 and, therefore, suffers the same deficiency as Claim 19.

Claim Rejections - 35 USC § 101

8. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

9. **Claims 6-20 and 24-26** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The result of **Claims 6-20 and 24-26** is directed to the act of “reading,” which does not appear to be a tangible result so as to constitute a practical application of the idea. The act of “reading” is merely a thought or an abstract idea and does not appear to produce a tangible result even if the step of “reading” does occur, since the result of that reading is not conveyed in the real world. The result is a reading, which is neither used in a disclosed practical application nor made available for use in a disclosed practical application. It also does not appear that the usefulness of the reading can be realized from the claimed steps to support a disclosed specific, substantial, and credible utility so as to produce a useful result.

Therefore, the claims do not meet the statutory requirement of 35 U.S.C. § 101, since the claims are not directed to a practical application of the § 101 judicial exception producing a result tied to the physical world.

Claim Rejections - 35 USC § 102

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

11. **Claims 1-11, 17-20, and 24-26** are rejected under 35 U.S.C. 102(e) as being anticipated by **Gungabeesoon (US 7,007,278)**.

As per **Claim 1**, Gungabeesoon discloses:

- a memory block (*see Figure 1: 102*);
- a COBOL program communicating with the memory block (*see Figure 1: 122*;

Column 11: 23-27, "... it is to be understood that the architecture but could also support legacy applications written in COBOL ... ");

- a socket (*see Figure 6: 626A and 626B*); and
- a COBOL routine callable from the COBOL program, the COBOL routine operable to read information from the socket and write the information read from the socket to the memory block in response to the COBOL program call (*see Figure 6; Column 4: 53-58, "Operating system 120 and applications 122 reside in memory 102."; Column 10: 29-31, "The operating system application runtime 430 calls the Read_Data and Write_Data methods to redirect data flow between the application and the Internet."*).

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Gungabeesoon further discloses:

- wherein the COBOL program further communicates with the COBOL routine to initiate the COBOL routine communication with the socket and the memory block (*see Figure 6; Column 11: 8-10, "Subsequent interactions between the client interface on the network user agent 570 and the application 122 flows through the socket connections 626a and 626b."*).

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and Gungabeesoon further discloses:

- wherein the COBOL routine is further defined as a subroutine of the COBOL program (see Column 8: 14-17, "Each legacy application 122 has data 422 to be input/output to/from the application runtime operating system 430 according to the program I/O code 410 through the compiler runtime 420.").

As per **Claim 4**, the rejection of **Claim 1** is incorporated; and Gungabeesoon further discloses:

- wherein the COBOL routine is further defined as a COBOL library having a plurality of routines callable by the COBOL program (see Figure 6; Column 10: 37-67 through Column 11: 1-22).

As per **Claim 5**, the rejection of **Claim 1** is incorporated; and Gungabeesoon further discloses:

- wherein the COBOL routine is further defined as a compiler enabled function usable by the COBOL program (see Column 8: 14-17, "Each legacy application 122 has data 422 to be input/output to/from the application runtime operating system 430 according to the program I/O code 410 through the compiler runtime 420.").

As per **Claim 6**, Gungabeesoon discloses:

Art Unit: 2191

- requesting, by a COBOL program, information from a socket (*see Column 11: 13-18, "The input data is then forwarded to socket or queue 626a as in step 642c, to the other application socket or queue 626b and I/O buffers if any and to the application runtime component 430, and eventually to the legacy program 122 that was waiting on a Read_Data method 640b."*);
- retrieving, by a COBOL routine, information from the socket (*see Column 11: 13-18, "The input data is then forwarded to socket or queue 626a as in step 642c, to the other application socket or queue 626b and I/O buffers if any and to the application runtime component 430, and eventually to the legacy program 122 that was waiting on a Read_Data method 640b."*);
- writing, by the COBOL routine, information read from the socket to a memory block (*see Column 4: 53-58, "Memory 102 is a random-access semiconductor memory for storing data and programs ... Operating system 120 and applications 122 reside in memory 102."*); and
- reading from the memory block, by the COBOL program, the information (*see Column 4: 53-58, "Memory 102 is a random-access semiconductor memory for storing data and programs ... Operating system 120 and applications 122 reside in memory 102."*).

As per **Claim 7**, the rejection of **Claim 6** is incorporated; and Gungabeesoon further discloses:

- managing, by the COBOL routine, a connection with the socket (*see Column 10: 51-54, "... the application invoker 660 which creates a socket 626b and makes a connection to the socket in the network server process as in step 620."*).

As per **Claim 8**, the rejection of **Claim 7** is incorporated; and Gungabeesoon further discloses:

- wherein managing includes listening on the socket connection (*see Column 10: 54-57, "After a connection is established in step 622 between the two sockets 626a and 626b, the socket 626b in the network server process waits for data from the legacy program."*).

As per **Claim 9**, the rejection of **Claim 7** is incorporated; and Gungabeesoon further discloses:

- wherein managing includes disconnecting the connection with the socket (*see Column 11: 20-22, "When the application ends, control returns to the application invoker 660 which closes the endpoint connection as in step 620."*).

As per **Claim 10**, the rejection of **Claim 6** is incorporated; and Gungabeesoon further discloses:

- establishing, by the COBOL routine, a connection with the socket (*see Column 10: 51-54, "... the application invoker 660 which creates a socket 626b and makes a connection to the socket in the network server process as in step 620."*).

As per **Claim 11**, the rejection of **Claim 10** is incorporated; and Gungabeesoon further discloses:

Art Unit: 2191

- wherein the connection with the socket is established in response to a request from the COBOL program (*see Column 10: 45-50, "The servlet instance 610, upon receiving the XML or HTTP request for invocation, at step 616 creates a socket 626a ..."*).

As per **Claim 17**, the rejection of **Claim 6** is incorporated; and Gungabeesoon further discloses:

- wherein the COBOL routine further includes a coordination module to coordinate such that the COBOL routine only reads when the socket has information and only writes when the socket is not full (*see Column 10: 57-67 through Column 11: 1-7, "The application invoker 660 writes the descriptor of the application-side socket 626b to an environment variable, activates data redirection through an application programming interface and invokes the legacy application 122, as shown in step 624."*).

As per **Claim 18**, the rejection of **Claim 6** is incorporated; and Gungabeesoon further discloses:

- initiating a call to an operating system by the COBOL routine to establish a socket connection (*see Figure 2: 221; Figure 6; Column 10: 8-17, "For outbound data the servlet instance 610 accepts a data buffer from the socket, looks up the record identifier of the data, instantiates, and populates the associated data object, e.g., the JavaBean, and activate the associated JavaServer Page to serve the data to the network user agent for display. For inbound data, the servlet instance 610 packages the data into a data buffer and submits it to the application via the socket and Publish-to-Web runtime component."*).

As per **Claim 19**, the rejection of **Claim 18** is incorporated; and Gungabeesoon further discloses:

- wherein the call to an operating system is further defined as a bit-level call to an operating system of a mainframe computer system (*see Figure 2: 221; Figure 6; Column 10: 8-17, "For outbound data the servlet instance 610 accepts a data buffer from the socket, looks up the record identifier of the data, instantiates, and populates the associated data object, e.g., the JavaBean, and activate the associated JavaServer Page to serve the data to the network user agent for display. For inbound data, the servlet instance 610 packages the data into a data buffer and submits it to the application via the socket and Publish-to-Web runtime component."*).

As per **Claim 20**, the rejection of **Claim 19** is incorporated; and Gungabeesoon further discloses:

- wherein the COBOL routine is further defined as written in COBOL programming language (*see Figure 1: 122; Column 11: 23-27, "... it is to be understood that the architecture but could also support legacy applications written in COBOL ..."*).

As per **Claim 24**, Gungabeesoon discloses:

- reading, by a routine, information from a socket (*see Column 11: 13-18, "The input data is then forwarded to socket or queue 626a as in step 642c, to the other application socket or queue 626b and I/O buffers if any and to the application runtime component 430, and eventually to the legacy program 122 that was waiting on a Read_Data method 640b."*);

Art Unit: 2191

- writing, by the routine, the information to an area (*see Column 4: 53-58, "Memory 102 is a random-access semiconductor memory for storing data and programs ... Operating system 120 and applications 122 reside in memory 102."*); and
- reading, by a COBOL program, the information from the area, the COBOL program and the routine operating in the same runtime environment (*see Figure 6; Column 4: 53-58, "Memory 102 is a random-access semiconductor memory for storing data and programs ... Operating system 120 and applications 122 reside in memory 102."*).

As per **Claim 25**, the rejection of **Claim 24** is incorporated; and Gungabeesoon further discloses:

- wherein the area is a file (*see Column 11: 2-7, "The network page is populated with data from the data object as in step 652 ..."*).

As per **Claim 26**, the rejection of **Claim 24** is incorporated; and Gungabeesoon further discloses:

- wherein the area is a memory area (*see Figure 1: 102*).

Claim Rejections - 35 USC § 103

12. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

13. **Claims 12-14** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Gungabeesoon** (US 7,007,278) in view of **Vermeire et al.** (US 6,931,623).

As per **Claim 12**, the rejection of **Claim 6** is incorporated; however, **Gungabeesoon** does not disclose:

- wherein the COBOL routine provides an address to the COBOL program, the address identifying a location of the memory block where the information is written.

Vermeire et al. disclose:

- wherein the COBOL routine provides an address to the COBOL program, the address identifying a location of the memory block where the information is written (*see Column 4: 35-44, "... a reference to the binary data contained within the record layout at the time the programming call to read or write data. The reference to the binary data is most likely a memory address (a "pointer") as implemented in most programming languages."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of **Vermeire et al.** into the teaching of **Gungabeesoon** to include wherein the COBOL routine provides an address to the COBOL

Art Unit: 2191

program, the address identifying a location of the memory block where the information is written. The modification would be obvious because one of ordinary skill in the art would be motivated to locate data in memory.

As per **Claim 13**, the rejection of **Claim 12** is incorporated; however, Gungabeesoon does not disclose:

- mapping, by the COBOL program, the memory block into the COBOL program.

Vermeire et al. disclose:

- mapping, by the COBOL program, the memory block into the COBOL program (*see Column 6: 43-55, "An existing COBOL copybook, an example of which is shown in FIG. 3, or a PL/I record definition in the source code of an existing legacy application are examples of a source record definition." and "The source record definition is processed by a lexical analyzer FIG. 2 capable of translating the language-specific representation of a record layout into a language-neutral and computer-architecture neutral representation of the data layout ("metadata"). This metadata is stored on a persistent storage medium 35 of FIG. 12 and accessed and managed via the workbench. "*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Vermeire et al. into the teaching of Gungabeesoon to include mapping, by the COBOL program, the memory block into the COBOL program. The modification would be obvious because one of ordinary skill in the art would be motivated to locate data in memory.

Art Unit: 2191

As per **Claim 14**, the rejection of **Claim 13** is incorporated; however, Gungabeesoon does not disclose:

- wherein the mapping is accomplished using a copybook.

Vermeire et al. disclose:

- wherein the mapping is accomplished using a copybook (*see Column 6: 43-55, "An existing COBOL copybook, an example of which is shown in FIG. 3, or a PL/I record definition in the source code of an existing legacy application are examples of a source record definition."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Vermeire et al. into the teaching of Gungabeesoon to include wherein the mapping is accomplished using a copybook. The modification would be obvious because one of ordinary skill in the art would be motivated to describe the physical layout of data.

14. **Claims 15 and 16** are rejected under 35 U.S.C. 103(a) as being unpatentable over Gungabeesoon (US 7,007,278) in view of Ahmad et al. (US 5,745,748).

As per **Claim 15**, the rejection of **Claim 6** is incorporated; however, Gungabeesoon does not disclose:

- wherein the information is provided in an EBCDIC format and wherein the method further comprises converting the information from the EBCDIC format to an ASCII format.

Ahmad et al. disclose:

Art Unit: 2191

- wherein the information is provided in an EBCDIC format and wherein the method further comprises converting the information from the EBCDIC format to an ASCII format (*see Column 3: 18-21, "... if the data to be downloaded are in the EBCDIC format, as is common for mainframe computers, it must often be converted to the ASCII format for PC storage or use."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Ahmad et al. into the teaching of Gungabeesoon to include wherein the information is provided in an EBCDIC format and wherein the method further comprises converting the information from the EBCDIC format to an ASCII format. The modification would be obvious because one of ordinary skill in the art would be motivated to store or use the information in a PC (*see Ahmad et al. – Column 3: 18-21*).

As per **Claim 16**, the rejection of **Claim 15** is incorporated; however, Gungabeesoon does not disclose:

- wherein the conversion is accomplished by the COBOL routine.

Ahmad et al. disclose:

- wherein the conversion is accomplished by the COBOL routine (*see Column 3: 52-56, "... a system and method were needed to enable a mainframe-class application program under development in a PC-based COBOL development system to directly access data on a mainframe computer to which the PC was electronically linked."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Ahmad et al. into the teaching of Gungabeesoon to include wherein the conversion is accomplished by the COBOL routine. The

Art Unit: 2191

modification would be obvious because one of ordinary skill in the art would be motivated to perform the conversion to allow access to mainframe computer data (*see Ahmad et al.* – Column 3: 18-21).

15. **Claims 21-23** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Gungabeesoon** (US 7,007,278).

As per **Claim 21**, Gungabeesoon discloses:

- a memory block (*see Figure 1: 102*);
- a COBOL program communicating with the memory block (*see Figure 1: 122*;

Column 11: 23-27, "... it is to be understood that the architecture but could also support legacy applications written in COBOL ... "); and

- a COBOL routine callable from the COBOL program, the COBOL routine operable to read information and write the information to the memory block in response to the COBOL program call (*see Figure 6; Column 4: 53-58, "Operating system 120 and applications 122 reside in memory 102."; Column 10: 29-31, "The operating system application runtime 430 calls the Read_Data and Write_Data methods to redirect data flow between the application and the Internet."*).

However, Gungabeesoon does not disclose:

- a pipe.

Official Notice is taken that it is old and well known within the computing art to utilize a pipe. Pipelines are often implemented in a multitasking operating system, by chaining processing

Art Unit: 2191

elements (processes, threads, etc.). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a pipe. The modification would be obvious because one of ordinary skill in the art would be motivated to facilitate communications within the same computer.

As per **Claim 22**, the rejection of **Claim 21** is incorporated; and Gungabeesoon further discloses:

- wherein the memory block is further defined as a mainframe memory block and wherein the COBOL program and COBOL routine are operable on a mainframe computer system (*see Figure 2: 202; Figure 6; Column 5: 45-46, "FIG. 2 is an example of a network server 200 which may access a legacy application stored on the computer 100."*).

As per **Claim 23**, the rejection of **Claim 22** is incorporated; and Gungabeesoon further discloses:

- a create module communicating with a computer system and operable to create a pipe connection (*see Figure 6; Column 10: 51-54, "... the application invoker 660 which creates a socket 626b and makes a connection to the socket in the network server process as in step 620."*);
- a connect module operable to promote attachment to the pipe connection (*see Figure 6; Column 10: 51-54, "... the application invoker 660 which creates a socket 626b and makes a connection to the socket in the network server process as in step 620."*);

Art Unit: 2191

- an open module operable to open the pipe connection to promote communication via the pipe connection (*see Figure 6; Column 10: 51-54, "... the application invoker 660 which creates a socket 626b and makes a connection to the socket in the network server process as in step 620."*);

- a write module operable to write information to the pipe connection, the write module operable to verify that the pipe connection is not full prior to writing information and further operable to block when the pipe connection is full (*see Figure 6; Column 10: 62-66, "When the legacy application 122 reaches an I/O instruction, output data is sent as in step 632 to the application runtime component 430 of the computer's operating system which calls the Write_Data method as in 640a to redirect data to the application-side socket 626b."*);

- a read module coupleable to the pipe connection to read information from the pipe connection (*see Figure 6; Column 11: 13-18, "The input data is then forwarded to socket or queue 626a as in step 642c, to the other application socket or queue 626b and I/O buffers if any and to the application runtime component 430, and eventually to the legacy program 122 that was waiting on a Read_Data method 640b."*);

- a release module to release the pipe connection (*see Figure 6; Column 11: 20-22, "When the application ends, control returns to the application invoker 660 which closes the endpoint connection as in step 620."*);

- a remove module to remove the pipe connection from the computer system (*see Figure 6; Column 11: 20-22, "When the application ends, control returns to the application invoker 660 which closes the endpoint connection as in step 620."*); and

Art Unit: 2191

- a delete module to delete the pipe connection wherein the pipe connection is closed
(see Figure 6; Column 11: 20-22, "When the application ends, control returns to the application invoker 660 which closes the endpoint connection as in step 620. ").

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR

Art Unit: 2191

system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QC / *ac*
April 10, 2007



WEI ZHEN
SUPERVISORY PATENT EXAMINER